CSE 214 – Data Structures

Homework 1 – Spring 2019

TA to contact: Avik Kadakia avik.kadakia@stonybrook.edu

Homework 1 – due **Thursday, February 21th no later than 11:59 PM**

Imagine you work for the Federal Department of Health and Human Services and you are their lead data manager. One day the Secretary of HSS comes to you with a .csv file of records of some people and wants you to store them in the system. The .csv file contains the names of some citizens, their sex, age, height in inches, and weight in pounds. You have to read the .csv file and store all the data in an Array (DO NOT use an ArrayList). The .csv file is provided to you with the pdf on blackboard. The one provided to you is different from the ones we will use for testing!

The file is sorted by name. You have to read the file, print after storing it and giving options to manipulate it, and lastly, save it to a .csv type file to be processed again. Make a program that can run on its own and does not crash. Use exception handling as required. If the program does not compile, you will receive a 0 for it. You have to write all the CODE BY YOURSELF. Any form of plagiarism will result in 0 in the homework and possibly even a Q for the course.

Required Classes:

The following classes are required for this assignment. Each class provides a description and the specifications necessary to complete each class. If you feel that additional methods or variables would be useful, feel free to add them during your implementation as you see fit. However, all the variables and methods in the following specifications must be included in your project.

1. Person

Write a fully documented class named Person **that contains biological statistics of each person in the .csv file provided. The** Person **class should contain variables for the person's name, age, gender, height, and weight. In addition, it should also implement the** toString() **method, which should print all of the person's data in tabular form.**

- public Person() – **constructor (You may include a constructor with parameter)**
- **One** int **variables:**
  - age
- **Two** double **variables:**
  - height
  - weight
- **Two** String **variables:**
  - name

    - gender
- public String toString() **- returns string of data members in tabular form.**
- **Getters and setters for all members.**

2. **PersonDataManager**

Write a fully documented class named PersonDataManager **that reads the .csv file and creates** Person **objects that are later stored in an Array. In addition to that, you should print the data from the array as a table.**

- **An Array:** Person[] people; (Base the size on the number of people in the document to be read)

3. public static PersonDataManager buildFromFile (String location) throws IllegalArgumentException
    - ***Brief:***
        - **Uses the File class to read the .csv file and store the information in the array created above (**Person[] people**).**
    - ***Parameters:***
        - location – File path **of the file to be read as a** String**.**
    - ***Preconditions:***
        - **None.**
    - ***Postconditions:***
        - **None.**
    - ***Returns:***
        - PersonDataManager **contructed from the .csv file.**
    - ***Throws:***
        - IllegalArgumentException**: Thrown if the data entered is in the wrong format. For example, there is a letter in height or a number in the *person's* name.**

- public void addPerson (Person newPerson) throws PersonAlreadyExistsException
    - ***Brief:***
        - **Adds the** Person **object in the data structure chosen in the correct alphabetical order that the list is in. If the array you are using is entirely full, create a new array with more space and copy the elements in the other array to this array. Use that bigger array as the new array.**
    - ***Parameters:***
        - newPerson – Person **object to be added.**
    - ***Preconditions:***
        - **The person does not exist in the list.**
    - ***Postconditions:***

        - **The person is added to the list.**
    - ***Returns:***
        - **None.**
    - ***Throws:***
        - PersonAlreadyExistsException**: Thrown if a person with all the same biological statistics already exists in the list.**
- public void getPerson (String name) throws PersonDoesNotExistsException
    - ***Brief:***
        - **Retrieves and prints the data of the** Person **object from the data structure chosen.**
    - ***Parameters:***
        - name – **Name of the** Person **object to be printed.**
    - ***Preconditions:***
        - **The person with the given name exists.**
    - ***Postconditions:***
        - **None.**
    - ***Returns:***
        - **None.**
    - ***Throws:***
        - PersonDoesNotExistsException**: Thrown if a person with the given name does not exist.**
- public void removePerson (String name) throws PersonDoesNotExistsException
    - ***Brief:***
        - **Removes the** Person **from the data structure chosen. Once the person is removed, move all the people after that location in the array to one space left.**
    - ***Parameters:***
        - name – **Name of the** Person **object to be removed.**
    - ***Preconditions:***
        - **The person with the given name exists.**
    - ***Postconditions:***
        - **None.**
    - ***Returns:***
        - **None.**
    - ***Throws:***
        - PersonDoesNotExistsException**: Thrown if a person with the given name does not exist.**
- public void printTable()
    - ***Brief:***

        - **Prints the** PersonDataManager **in tabular form.**

4. **PersonManager**

Write a fully documented class named PersonManager**. This class will allow the user to interact with the database by listing the people, adding to the list, removing, and retrieving people from the list.**

- **A** PersonDataManager **variable:**
    - personDataManager
- public static void main (String [] args)
    - ***Brief:***
        - **This method should implement the following menu options:**

- **(I) – Import from File**
- **(A) – Add Person**
- **(R) – Remove Person**
- **(G) – Get Info on Person**
- **(P) – Print Table**
- **(S) – Save to File**
- **(Q) – Quit**

Sample Input / Output

// Comment in green, input in red, output in black

Starting...
Menu:

(I) – Import from File

(A) – Add Person

(R) – Remove Person

(G) – Get Information on Person

(P) – Print Table

(S) – Save to File

(Q) – Quit

```
Please select an option: I
Please enter a location: biostats.csv // provided by the user

Loading...
Person data loaded successfully!
// Menu not printed in sample i/o but you should print it again.

Please select an option: P
    Name    |     Age     |    Gender    |     Height      |    Weight
=====================================================================
    Alex    |     41      |      M       | 6 feet 2 inches  | 170 pounds
    Bert    |     42      |      M       | 5 feet 8 inches  | 166 pounds
    Carl    |     32      |      M       | 5 feet 10 inches | 155 pounds
    Dave    |     39      |      M       | 6 feet 0 inches  | 167 pounds
    Elly    |     30      |      F       | 5 feet 6 inches  | 124 pounds
    Fran    |     33      |      F       | 5 feet 6 inches  | 115 pounds
    Gwen    |     26      |      F       | 5 feet 4 inches  | 121 pounds
    Hank    |     30      |      M       | 5 feet 11 inches | 158 pounds
    Ivan    |     53      |      M       | 6 feet 0 inches  | 175 pounds
    Jake    |     32      |      M       | 5 feet 9 inches  | 143 pounds
    Kate    |     47      |      F       | 5 feet 9 inches  | 139 pounds
    Luke    |     34      |      M       | 6 feet 0 inches  | 163 pounds
    Myra    |     23      |      F       | 5 feet 2 inches  |  98 pounds
    Neil    |     36      |      M       | 6 feet 3 inches  | 160 pounds
    Omar    |     38      |      M       | 5 feet 10 inches | 145 pounds
    Page    |     31      |      F       | 5 feet 7 inches  | 135 pounds
    Quin    |     29      |      M       | 5 feet 11 inches | 176 pounds
    Ruth    |     28      |      F       | 5 feet 5 inches  | 131 pounds

// Menu not printed in sample i/o but you should print it again.

Please select an option: G

Please enter the name of the person: Ivan

Ivan is a 53 year old male who is 6 feet and 0 inches tall and weighs 175 pounds.

// Menu not printed in sample i/o but you should print it again.

Please select an option: A

Please enter the name of the person: John

Please enter the age: 30

Please enter the height (M or F): M
Please enter the height (in inches): 71

Please enter the weight (in lbs): 150

John has been added to the list!
```

// Menu not printed in sample i/o but you should print it again.

```
Please select an option: P

   Name   |   Age   |   Gender   |     Height      |    Weight
===================================================================
   Alex   |   41    |     M      |  6 feet 2 inches |  170 pounds
   Bert   |   42    |     M      |  5 feet 8 inches |  166 pounds
   Carl   |   32    |     M      | 5 feet 10 inches |  155 pounds
   Dave   |   39    |     M      |  6 feet 0 inches |  167 pounds
   Elly   |   30    |     F      |  5 feet 6 inches |  124 pounds
   Fran   |   33    |     F      |  5 feet 6 inches |  115 pounds
   Gwen   |   26    |     F      |  5 feet 4 inches |  121 pounds
   Hank   |   30    |     M      | 5 feet 11 inches |  158 pounds
   Ivan   |   53    |     M      |  6 feet 0 inches |  175 pounds
   Jake   |   32    |     M      |  5 feet 9 inches |  143 pounds
   John   |   30    |     M      | 5 feet 11 inches |  150 pounds
   Kate   |   47    |     F      |  5 feet 9 inches |  139 pounds
   Luke   |   34    |     M      |  6 feet 0 inches |  163 pounds
   Myra   |   23    |     F      |  5 feet 2 inches |   98 pounds
   Neil   |   36    |     M      |  6 feet 3 inches |  160 pounds
   Omar   |   38    |     M      | 5 feet 10 inches |  145 pounds
   Page   |   31    |     F      |  5 feet 7 inches |  135 pounds
   Quin   |   29    |     M      | 5 feet 11 inches |  176 pounds
   Ruth   |   28    |     F      |  5 feet 5 inches |  131 pounds
```

// Menu not printed in sample i/o but you should print it again.

```
Please select an option: R

Please enter the name of the person: Ruth

Ruth has been removed!
```

// Menu not printed in sample i/o but you should print it again.

```
Please select an option: P

   Name   |   Age   |   Gender   |     Height      |    Weight
===================================================================
```

Alex	41	M	6 feet 2 inches	170 pounds
Bert	42	M	5 feet 8 inches	166 pounds
Carl	32	M	5 feet 10 inches	155 pounds
Dave	39	M	6 feet 0 inches	167 pounds
Elly	30	F	5 feet 6 inches	124 pounds
Fran	33	F	5 feet 6 inches	115 pounds
Gwen	26	F	5 feet 4 inches	121 pounds
Hank	30	M	5 feet 11 inches	158 pounds
Ivan	53	M	6 feet 0 inches	175 pounds
Jake	32	M	5 feet 9 inches	143 pounds
John	30	M	5 feet 11 inches	150 pounds
Kate	47	F	5 feet 9 inches	139 pounds
Luke	34	M	6 feet 0 inches	163 pounds
Myra	23	F	5 feet 2 inches	98 pounds
Neil	36	M	6 feet 3 inches	160 pounds
Omar	38	M	5 feet 10 inches	145 pounds
Page	31	F	5 feet 7 inches	135 pounds
Quin	29	M	5 feet 11 inches	176 pounds

// Menu not printed in sample i/o but you should print it again.

Please select an option: S

Please select a name for the file: newBioStats.csv

A file named newBioStats.csv has been created!

// Menu not printed in sample i/o but you should print it again.

Please select an option: A

Please enter the name of the person: Janet

Please enter the age: B0

The input you entered is incorrect. Please try again!

// Menu not printed in sample i/o but you should print it again.

Please select an option: Q

Sorry to see you go!